Weekly report

Introduction

The presented work will be introduced in three aspects below. Firstly, the review paper about multivariate spatial data visualization has been modified as Prof.Chen told me. The second is related to the vis paper collection. Currently, all the resources from pacific visualization 2013 are uploaded to the FTP and SVN for the internal access, and the external access could be achieved via the link for the public version. The last aspect refers to the research work. The project is considered to be a parallel one and the data structure is modified to enhance the performance. Besides that, paper reading is still kept going as a usual work. In the second section, I will give the details of last two jobs and mainly focus on the third one.

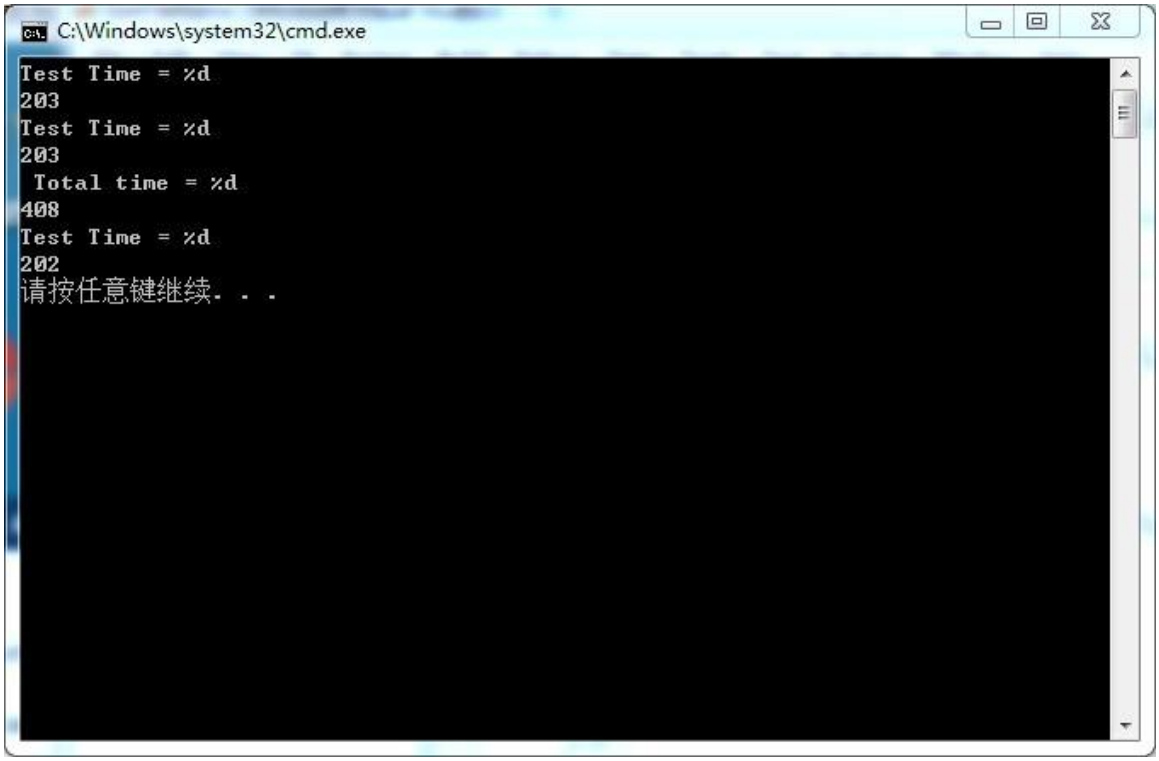
Details

As for the performance, the original version is too slow in the term of compute speed and memory construct is bad (too much new and delete operation, because the two dimension array is the main data structure).

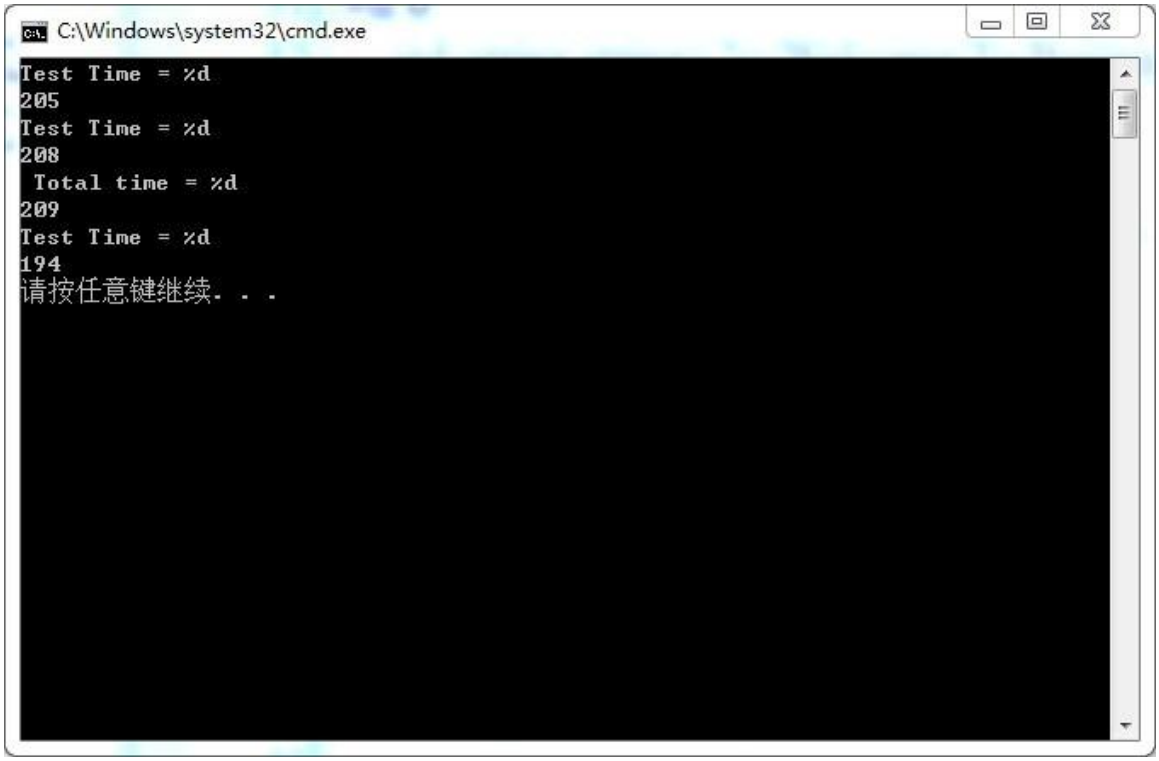
Considered the expansion, the convenient of test and the performance, I change the main data structure from the two dimension array([number of vector][dimension of vector]) to one dimension array. This operation throws almost new and delete operations away, thus enhancing the speed of memory allocation. For example, if we use the old version, when we test for a 256*256*256 volume dataset(suppose the block size is 1*1*1), the corresponding memory allocation should include 256*256*256 new and delete operations while just only one new and delete operation in the current time.

After the update of the whole project, I speed up the computation of codebook. Here, I use OpenMP library as the method to accelerate and it is multi-core approach. My computer is 8 threads, 4 cores, the acceleration show could be seen as the pictures depicted below. The test loop code is showed as following code segment and we do the test operation twice. Picture 1 and 2 are the corresponding result for the serial and parallel code. Pay attention to the “Total time” item, in the serial version, the total time is nearly the sum of the two operation while it equals to only just one test time in the parallel version(here ,we used two threads).

```
void test()
{
    int a = 0;
    clock_t t1 = clock();
    for( int i = 0; i < 100000000; i ++ )
    {
        a += i;
    }
    clock_t t2 = clock();
    std::cout<<"Test Time = \n"<<t2-t1<<std::endl;
}
```



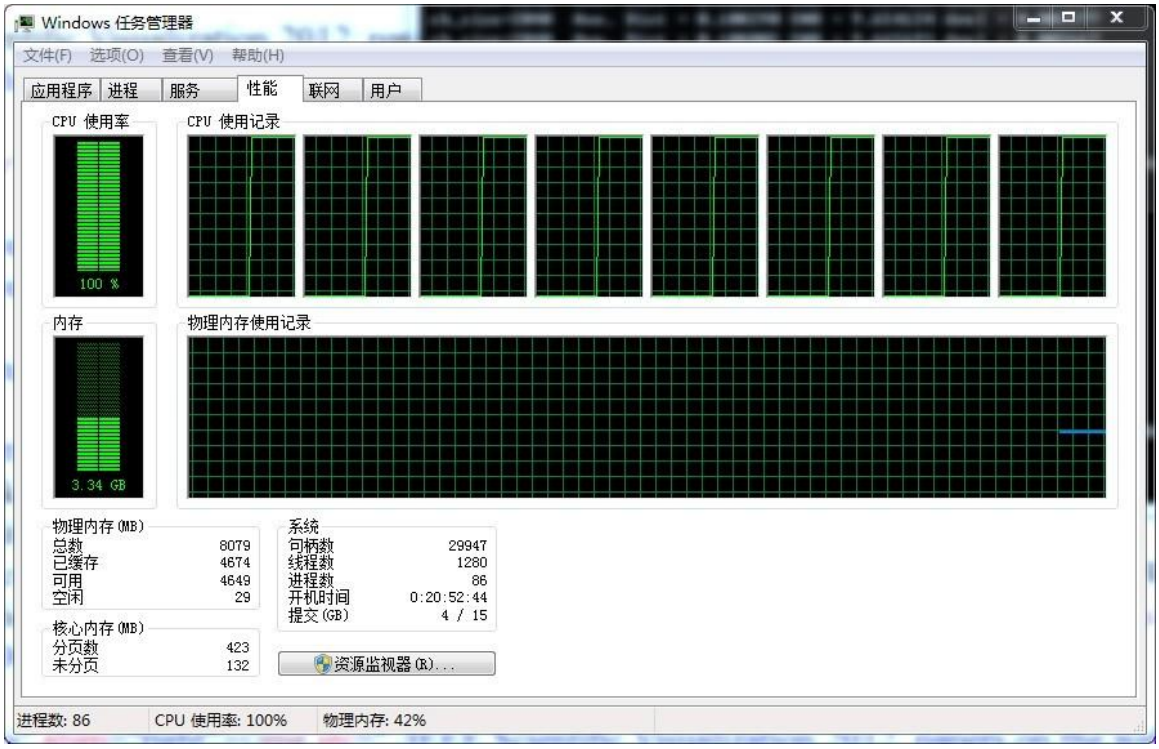
Picture 1



Picture2

In the real compression code, I use 8 threads and the memory use and CPU use situation is showed as picture 3. CPU is full now and the compute time of codebook is 4 times faster than the serial code in my machine. For example, for the 256*256*256 size volume data, it takes me 4 hours now while it is 17hours before. If we use Dr.Zhu's PC, it will only cost 50mins.

As for the paper reading, Xiaofei He's work and the ones related to it are reviewed in the last week. We will have a discussion nest week with Haidong Chen and Biao Zhu.



Picture3

Conclusion and Future Work

The most important work that has been done last week is the code update, to the parallel version. It takes some time to improve it and verifies the result.
In the next week, I will get all the compute results for all the datasets, including the ones from others' paper.

The patent writing will be done these two days.
The climate data extraction will be done this week, as the prepare for the demo.
If I still have time, EuroVis paper collection will in the list.

Reference

1.《多核计算与程序设计》 周伟明
2. <A Unified Active and Semi-Supervised Learning Framework for Image Compression>
3. <Learning to Compress Images and Videos>
4. <Fast, Accurate Detection of 100,100 Object Classes on a Single Machine>